Middle East Technical University - Department of Computer Engineering

# CENG 232

Logic Design

Spring 2019-2020

Quiz 2

**Due date:** Thursday, April 2, 2020, 22:30

## 1 Quiz Rules

**You can ask your questions using chat functionality of the odtuclass. You can reach the chat from here. This is an online chat system provided by odtuclass. However, if you leave or close the page, the session will be saved to previous sessions and you will not be able to catch the ongoing conversation.**
Your Logisim quiz consists of 3 questions. Labelling conventions are provided in each question section.

- Please implement each question in different circ file.
- "Encoder" question's circ file should be named as e{Student number}Encoder.circ
  **Example : e1234567Encoder.circ**
- "Demultiplexer" question's circ file should be named as e{Student number}Demux.circ
  **Example : e1234567Demux.circ**
- "Register" question's circ file should be named as e{Student number}Register.circ
  **Example : e1234567Register.circ**

## 2 Encoder

Encoder circuitry outputs the binary representation of the activated input hence yields reverse function of Decoder circuitry. You are expected to implement 4x2 encoder. This encoder consists of 4 inputs (I3, I2, I1, I0) and 2 outputs (O1, O0). At a given time only one of the inputs will be activated (1).

| Input | | | | Output | |
|---|---|---|---|---|---|
| I3 | I2 | I1 | I0 | O1 | O0 |
| 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 |

### 2.1 Labelling Specifications

- Your input pins should be labelled as I3, I2, I1, I0.
- Your output pins should be labelled as O1, O0
- Label properties are case-sensitive. **Please be very careful on the correct naming of labels.**
- You will receive grade **penalty** unless labelling is done properly.

# 3 Demultiplexer

Demultiplexer circuitry takes an input and transfers it to the selected output. It can be likened to a railway controller which connects a specific railway to different directions. You are expected to implement 1x4 demultiplexer consisting of 1 input (I0), 2 selector bits (S1, S0) and 4 outputs (O3, O2, O1, O0) by using **Controlled Buffer** component (In Logisim Ceng232 Gates → Controlled Buffer).

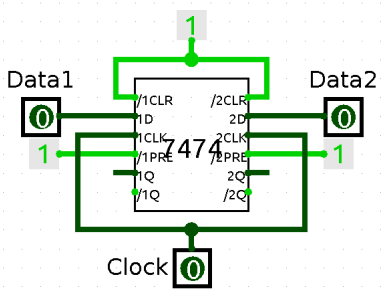
| Input | Selector | | Output | | | |
|---|---|---|---|---|---|---|
| I0 | S1 | S0 | O3 | O2 | O1 | O0 |
| A | 0 | 0 | 0 | 0 | 0 | A |
| A | 0 | 1 | 0 | 0 | A | 0 |
| A | 1 | 0 | 0 | A | 0 | 0 |
| A | 1 | 1 | A | 0 | 0 | 0 |

Where A $\in \{0, 1\}$.

## 3.1 Labelling Specifications

- Your input pins should be labelled as I0, S1, S0.
- Your output pins should be labelled as O3, O2, O1, O0
- Label properties are case-sensitive. **Please be very careful on the correct naming of labels.**
- You will receive grade **penalty** unless labelling is done properly.

# 4 Register




The figure here displays Logisim schematic of dual data type positive-edge trigger flip-flop IC (74LS74). IC contains 2 separate D-Flip Flops on a single package. Each flip-flop has /CLR (clear), D (data), CLK (clock), /PRE (preset), Q and /Q pins. When a rising edge (CLK=0 → CLK=1) occurs on CLK pin and /PRE pin is connected value of 1, the value presented on D pin is stored in the flip-flop and transferred to Q. Moreover, /Q outputs the complement value of pin Q. When a 0 value is supplied to /CLR pin, immediately value of 0 is stored in flip-flop without requiring a rising edge on CLK. A similar behaviour is observed when /PRE is connected to 0 but this time value of 1 is stored immediately rather than 0. /CLR and /PRE pins are active low pins. In order to be able to use flip-flop in normal operation, /CLR and /PRE pins should be connected to value of 1.
Suppose that we would like to implement a simple 8-bit register (array of flip-flops) and its controller. This controller should be capable of changing value of any bit in this register. This controller has a clock (CLK), data (D) and three-bit address pins (A2, A1, A0). Address pins select the bit to be altered with value on D pin. When a rising edge occurs on CLK pin the value on D is transferred to target bit of the register and the other bits remain unchanged.

Table illustrates **a sample behaviour** of memory controller and its consequence on register bits for a given time interval. Before any operation is applied, time step = 0, the bits of register store value of 0. At time step = 1, the controller selects Bit0 and receives a data bit of 1. When a rising edge occurs on CLK, it transfers value of 1 to Bit0. It stores value of 0 in Bit2 at time step 2. At time step 3 it stores value of 1 in Bit6 and so on.

| Time | Data | Address | | | CLOCK | Bits | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Step | D | A2 | A1 | A0 | CLK | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
| 1 | 1 | 0 | 0 | 0 | 0 → 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 2 | 0 | 0 | 1 | 0 | 0 → 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 3 | 1 | 1 | 1 | 0 | 0 → 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 4 | 1 | 0 | 1 | 1 | 0 → 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 5 | 0 | 0 | 0 | 0 | 0 → 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 6 | 1 | 1 | 1 | 1 | 0 → 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 7 | 0 | 0 | 1 | 1 | 0 → 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 1 | 1 | 0 | 0 | 0 → 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |

You are expect to implement this register and its controller in Logisim.

### 4.1 Labelling Specifications

- Your input pins should be labelled as D, CLK, A2, A1, A0.
- Your output pins should be labelled as Bit7, Bit6, Bit5, Bit4, Bit3, Bit2, Bit1, Bit0.
- You should use **constant 1** for /CLR and /PRE pins, their 0 values will not be tested.
- Label properties are case-sensitive. **Please be very careful on the correct naming of labels.**
- If you need to feed any input with a constant value, you can use a constant gate.
- You will receive grade **penalty** unless labelling is done properly.

## 5 Cheating Policy

All the work should be individual and there is zero tolerance policy for cheating. See the course website for further information about cheating policy.

## 6 References

CENG Logisim Version.